In the Claims:

Please amend the above-identified application as follows:

Claim 1 (previously amended) A method of designing logical circuits, comprising the steps of:

a. representing logic of a logical circuit to be designed as points and vectors in a vector space; and

b. using the points and vectors in said vector space to simplify the logic of the logical circuit to a simpler form; and

c. designing the logical circuit using the simpler form.

Claim 2 (previously amended) A method of manufacturing logical circuits, comprising the steps of:

a. representing logic of a logical circuit to be manufactured as points and vectors in a vector space; and

b. using the points and vectors in said vector space to simplify the logic of the logical circuit to a simpler form; and

c. using the simpler form to implement the logical circuit in hardware.

Claim 3 (original) A method of simplifying logical circuits, comprising the steps of:

a. representing the logic of a logical circuit as points and vectors in a vector space; and

b. modifying the representation in vector space using at least one process rule of a set of process rules to simplify the logic.

Claim 4 (currently amended) The method of claim 3 in which at least one process rule of a set of process rules consist~~ings~~ of one of the following process rules:

a. Process Rule 1– comprising the steps of

Representing in alternational normal schema, a target schema **t**, as a set of vectors in ANS-space,

Representing each clause or disjunct of **t** as a position vector pointing to **O** with **O** at one corner of a set of parallelograms made of propositional addresses to an **i**-point at another corner, and

Using any two other outside vertices of such a parallelogram as implicants which are among original clauses of **t**; and

b. Process Rule 2– comprising the steps of

Picking any two clauses, and

If there is a propositional address σ at a midpoint between component clauses, a vector from **i** to σ, is a simplification of and can replace relevant clauses; and

c. Process Rule 3—comprising the steps of

Generating **i**-implicants until each clause or vector has been used at least once, and

If a disjunct **d** of **t** cannot be used because it forms no propositional address with any other disjunct, then **d** must appear unmodified in a final schema which is the simplification of **t**; and

d. Process Rule 4—comprising the step of

If an **i**-point exists in **t**, deleting the vectors which produce it in favor of the

vector from **i** to **O**; and

e. Process Rule 5– comprising the step of

For a clause in a schema which subsumes another clause eliminating said clause which subsumes; and

f. Process Rule 6– comprising the step of

Couples ~~such as~~ in the form of **pq** v $\bar{p}\,\bar{q}$ or $\bar{p}\,\bar{q}$ **s** v **pq** $\bar{s}$ cannot be summed to zero at the origin; and

g. Process Rule 7– comprising the steps of

Translating vectors if a corresponding σ-point exists for an **i**-point then σ is the simplification of **i**, and

Identifying any superpositions of parallel arrows in opposite directions as equivalences, and

For equivalences, (a) drop the longer clause at either end of any double-headed arrow, (b) drop ~~pairs, triples etc. of~~ double-headed arrows which meet at a point in favor of the vector from that point to **O** and (c) drop a vector or clause in the target schema which is itself ~~the~~a resultant of any other two vectors; and

h. Process rule 8– comprising the steps of

Determining simplification is complete if in a representation of simplified layers which replaces the target schema no vectors or clauses are subsumed by others and no double-headed vectors remain.

Claim 5 (previously amended) Apparatus for simplifying logical circuits, comprising:

a. a processing element configured to represent logic of a logical circuit to be simplified as points and vectors in a vector space and to use the points and vectors to simplify the logic of the logical circuit to a simpler form.

Claim 6 (original) The apparatus of claim 5 in which the processing element is an optical computer.

Claim 7 (original) The apparatus of claim 5 in which the processing element is a digital computer.

Claim 8 (previously amended) The apparatus of claim 5 in which the processing element is an colorimetric computer.

Claim 9 (previously amended) The apparatus of claim 5 in which the processing element is an analog computer.

Claim 10 (previously amended) A computer program product, comprising:

a. a memory element; and

b. a computer program stored on said memory medium, said computer program comprising instructions for representing logic of a logical circuit to be designed as points and vectors in a vector space and for using the points and vectors in a vector space to simplify the logic of the logical circuit to a simpler form and for designing the logical circuit using the simpler form.

Claim 11 (previously amended) A computer program product, comprising:

a. a memory element; and

b. a computer program stored on said memory medium, said computer program comprising instructions for representing logic of a logical circuit to be manufactured as points and vectors in a vector space, and for using the points and vectors in a vector space to simplify the logic of the logical circuit to a simpler form, and for using the simpler form to implement the logical circuit in hardware.

Claim 12 (previously amended) A computer program product, comprising:

a. a memory element; and

b. a computer program stored on said memory medium, said computer program comprising instructions for representing logic of a logical circuit as points and vectors in a vector space, and for modifying the representation in a vector space using at least one process rule of a set of process rules to simplify the logic.